

---

# Learning from Multiple Outlooks

---

**Maayan Harel**

Department of Electrical Engineering, Technion, Haifa, Israel

MAAYANGA@TX.TECHNION.AC.IL

**Shie Mannor**

Department of Electrical Engineering, Technion, Haifa, Israel

SHIE@EE.TECHNION.AC.IL

## Abstract

We propose a novel problem formulation of learning a single task when the data are provided in different feature spaces. Each such space is called an outlook, and is assumed to contain both labeled and unlabeled data. The objective is to take advantage of the data from all the outlooks to better classify each of the outlooks. We devise an algorithm that computes optimal affine mappings from different outlooks to a target outlook by matching moments of the empirical distributions. We further derive a probabilistic interpretation of the resulting algorithm and a sample complexity bound indicating how many samples are needed to adequately find the mapping. We report the results of extensive experiments on activity recognition tasks that show the value of the proposed approach in boosting performance.

## 1. Introduction

It is often the case that a learning task relates to multiple representations, to which we refer as *outlooks*. Samples belonging to different outlooks may have varying feature representations and distinct distributions. Furthermore, the outlooks are not related through corresponding instances, but just by the common task.

Multiple outlooks may be found in many real life problems. For example, in activity recognition when data from different users, representing the outlooks, are collected from different sensors. Note that each outlook may have a totally different feature representations,

while the recognition task is common to all outlooks. The ability to learn from these different representations is formulated by multiple outlook learning. A different example for multiple outlooks learning is classification of document corpora written in different languages. In this case, each language represents a different outlook. In these situations, the transformations between the outlooks are unknown and feature or sample correspondence is not available. Consequently, it is rather difficult to learn the task at hand while exploiting the information in different representations.

The goal of multiple outlook learning is to use the information in all available outlooks to improve the learning performance of the task. We propose to approach this learning problem in a two step procedure. First, we map the empirical distributions of the different outlooks one to another. After the outlooks' distributions are matched, a generic classification algorithm can be applied using the available examples from all the outlooks.

This approach allows to transfer an outlook of which we have little information to another where we have more information. That is, mapping the data to the same space effectively enlarges our sample size and may also give us a better representation of the problem. We show that a classifier learned in the resulting space may outperform each single classifier.

In general, matching multiple distributions, without feature alignment or assuming a parametric model, is a difficult task. Therefore, we propose to match the empirical moments of the distributions as an approximation. We present an algorithm for finding one such mapping. The algorithm's objective is to find the optimal affine transformations of the outlooks' spaces, while maintaining isometry within classes. From a geometric point of view, our algorithm is based on matching the centers and the main directions of the outlooks' sample distributions. One virtue of the algorithm is its simple closed form solution.

---

Appearing in *Proceedings of the 28<sup>th</sup> International Conference on Machine Learning*, Bellevue, WA, USA, 2011. Copyright 2011 by the author(s)/owner(s).

## 2. Related work

Learning from multiple outlooks is related to other setups such as domain adaptation, multiple view learning and manifold alignment. The main challenge in these setups, as in ours, is that the training and test data are drawn from different distributions.

Domain adaptation tries to resolve a common scenario when some changes have been made to the test distribution, while the labeling function of the domains remains more or less the same. Some authors portray this situation by assuming a single hypothesis may classify both domains well (Blitzer et al., 2007), while others assume the target’s posterior probability is equal for the domains (Shimodair, 2000; Huang et al., 2007). The latter assumption is also referred to as the covariate shift problem.

Algorithms for domain adaptation may be roughly divided to three categories. One approach is to reweigh the training instances so they better resemble the test distribution (Shimodair, 2000; Huang et al., 2007). Such algorithms are derived from the covariate shift assumption, which is in some sense one of the outlook mapping goals. A different approach is to combine the classifiers learnt in each domain (Mansour et al., 2009). Last, some works suggest to change the feature representation of the domains. This may be carried out by choosing a subset of features (Satpal & Sarawagi, 2007), combination of features (Daumé III, 2007), or by finding some structural correspondence between features in different domains (Blitzer et al., 2006). All the described approaches entail an initial common feature representation for the domains. Thus domain adaptation is a special case of the multiple outlook problem, for the case of outlooks with a common feature space. In Section 6 we show that our approach can also be applied to this problem.

Multiple outlook learning is also closely related to the multi-view setup (Rüping & Scheffer, 2005). In this setup, each view contains the *same* set of samples represented by different features. Clearly, any multiple view data is also some instance of a multiple outlook data with the added requirement that each sample has observations from multiple outlooks. One common approach is to map a pattern matrix of each view to a consensus pattern by matching corresponding instances (Long et al., 2008; Hou et al., 2010). Note that in the multiple outlook framework each outlook contains a *unique* set of samples, thus sample to sample correspondence is impossible. Amini et al. (2009) considers the case when correspondence is missing for some instances, but assumes the existence of a mapping functions between the views.

Multi-view learning is sometimes referred to as manifold alignment. In manifold alignment we look for a transformation of two data sets with sample pairwise correspondence that minimizes the distance between them, in an unsupervised (Wang & Mahadevan, 2008) or a semi-supervised (Ham et al., 2005) manner. Wang & Mahadevan (2009) present manifold alignment without pairwise correspondence. To our knowledge, this is the only work on manifold alignment that does not assume a pairwise matching of the samples. The algorithm presented in this work is not originally suited for classification as our algorithm.

## 3. Mapping Two Outlooks

### 3.1. Problem Setting

The learner is given two outlooks belonging to separate input spaces  $\mathcal{X}_1$  and  $\mathcal{X}_2$  of dimension  $d^1$  and  $d^2$  respectively, with a common target  $\mathcal{Y} = \{1, \dots, c\}$ . We assume that all example pairs of a given outlook  $j = 1, 2$  are independently drawn from an unknown distribution  $\mathcal{D}_j$ , which is unique to each outlook. Denote by  $X_i^{(1)}$  and  $X_i^{(2)}$  the data matrices of class  $i$  of outlook 1 and 2, respectively. We use superscripts to denote the outlooks’ index, and subscripts to denote the classification class.

### 3.2. Multiple Outlook MAPping algorithm

In this section we present our main Multiple Outlook MAPping algorithm (MOMAP) for matching the representations of two outlooks. Throughout the derivations *outlook 2* is mapped to *outlook 1*, which is sometimes referred to as the final outlook. Our goal is to map an outlook where we have ample labeled data, to an outlook where little labeled information is available.

As a preliminary step to the mapping algorithm scaling is applied. The scaling is applied to each of the outlooks separately, and aims to normalize the features of all outlooks to the same range. Note that this stage may be done using unlabeled data when available.

Next, we use the labeled samples to match the two outlooks. The goal of this stage is to map the scaled representations by rotation and translation. Specifically, the mapping is performed by translating the means of each class to zero, rotating the classes to fit each other well, and then translating the means of the mapped outlook to the final outlook.

Let  $\left\{ \hat{\mu}_i^{(1)}, \hat{\mu}_i^{(2)} \right\}_{i=1}^c$  be the set of empirical means of the outlooks. We translate the empirical means of each

class of both outlooks to zero:

$$\hat{X}_i^{(j)} = X_i^{(j)} - \hat{\mu}_i^{(j)} \quad i = 1, \dots, c, j = 1, 2. \quad (1)$$

Next, we turn to matching the main directions of the classes by rotation. Note that a rotation matrix may be defined in many manners. We search for mappings in the set of all orthonormal matrices (rotation and reflection). Our choice of mapping by rotation is motivated by its isometry property, which allows us to maintain the relative distance between the samples. We construct utilization matrices for each of the outlooks as follows. Define  $D_i^{(j)}$  as the utilization matrix of outlook  $j$  and class  $i$ .  $D_i^{(1)}$  and  $D_i^{(2)}$  are concatenated matrices constructed from the  $h \leq \min(d^1, d^2)$  principal directions of the corresponding outlook and class. That is, the  $h$  eigenvectors of the empirical covariance matrices  $\hat{\Sigma}_i^{(1)}, \hat{\Sigma}_i^{(2)}$  corresponding to the  $h$  largest eigenvalues.

Using the utilization matrices we find the rotation matching the outlooks by solving the following optimization problem:

$$\{R_i\} = \arg \min_{\{R_i\}} \sum_{i=1}^c \left\| R_i D_i^{(2)} - D_i^{(1)} \right\|_F^2 \quad (2)$$

subject to:  $R_i^T R_i = I \quad i = 1, \dots, c,$

where  $\|\cdot\|_F$  is the Frobenius norm.

To gain some intuition on Problem (2) we disassemble a term in the sum of the objective function

$$\arg \min \left\| R_i D_i^{(2)} - D_i^{(1)} \right\|_F^2 = \arg \max \sum_{l=1}^h \mathbf{v}_{il}^{(1)T} R \mathbf{v}_{il}^{(2)},$$

where  $\mathbf{v}_{il}^{(j)}$  ( $l = 1, \dots, h$ ) are the principal directions of the  $i^{\text{th}}$  class of outlook  $j$ . We obtain that Problem (2) is equivalent to maximization of the sum of inner products between the principal directions of outlook 1 and the rotated principal directions of outlook 2, which in turn implies minimization of the first  $h$  principal angles between the classes of both outlooks.

Although Problem (2) is not convex it can be solved in closed form. For the solutions constructed in this stage we borrow techniques from the literature of Procrustes Analysis (Gower & Dijkstra, 2004). Problem (2) is equivalent to

$$\arg \max_{R_i} \sum_{i=1}^c \text{tr} \left( R_i D_i^{(2)} D_i^{(1)T} \right) \quad (3)$$

subject to:  $R_i^T R_i = I \quad i = 1, \dots, c.$

Problem (3) is separable, thus each component in the sum may be optimized separately. In the following derivations we drop the subscript  $i$  for brevity.

---

**Algorithm 1** *Matching two outlooks*


---

**Input:** empirical moments  $\hat{\mu}_i^{(j)} \forall i, j$ .  
**for**  $i = 1$  **to**  $c$  **do**  
 $\hat{X}_i^{(j)} = X_i^{(j)} - \hat{\mu}_i^{(j)} \quad j = 1, 2.$   
 $\tilde{X}_i^{(2)} = \text{MatchByRotation}(\hat{X}_i^{(1)}, \hat{X}_i^{(2)}).$   
 $X_{\text{Mapped}_i}^{(2)} = \tilde{X}_i^{(2)} + \hat{\mu}_i^{(1)}.$   
**end for**  
**Output:**  $X_{\text{Mapped}_i}^{(2)} \quad \forall i$

---



---

**Algorithm 2** *MatchByRotation*


---

**Input:** matrices  $\hat{X}^{(1)}, \hat{X}^{(2)}$ .  
 Construct matrices  $D^{(1)}, D^{(2)}$ .  
 Compute SVD factorization  $D^{(2)} D^{(1)T} = USV^T$ .  
 $R = VU^T$ .  
**Output:**  $\tilde{X}^{(2)} = \hat{X}^{(2)} R^T$ .

---

Let  $USV^T$  be the singular value decomposition (SVD) of  $D^{(2)} D^{(1)T}$ . Define  $Z = V^T R U$ . Then,

$$\text{tr} \left( R D^{(2)} D^{(1)T} \right) = \text{tr} \left( R U S V^T \right) =$$

$$\text{tr} (Z S) = \sum_{k=1}^d z_{kk} \sigma_k \leq \sum_{i=k}^d \sigma_k,$$

where  $\sigma_k$  is the  $k$ -th singular value of  $D^{(2)} D^{(1)T}$ . The upper bound is attained for  $R = VU^T$  since in that case  $Z = I$  (Algorithm 2).

After the rotation, we translate the classes to match the original means of the final outlook. The above derivation gives rise to an algorithm that matches two given outlooks. The algorithm is described in Algorithm 1.

*Remark 1.* Each outlook need not have the same dimension. In this case, the orthonormal constraint can not be obtained as  $R$  is no longer a square matrix. However, this problem can be easily solved. Suppose that  $D_i^{(1)}$  and  $D_i^{(2)}$  have different numbers of rows. Then, simply add rows of zeros to the smaller dimensional configuration until the dimensions are equalized. In this manner, we embed the smaller configuration in the space of the larger one.

*Remark 2.* Algorithm 1 does not rely on any corresponding instances in both outlooks. However, when available, such instances may aid the mapping accuracy and can be easily incorporated into the algorithm. It is possible to do so by adding columns of the corresponding instances to the utilization matrices.

## 4. Extension to Multiple outlooks

We present an extension of Algorithm 1 to the case of multiple outlooks. The multiple outlook scenario allows us to use the information available in all the outlooks to allow better learning of each one. To do so, we transform all the outlooks one to another. As for two outlooks, we begin by translating the means of each class of all the outlooks to zero. In the rotation step, the optimal rotations are found by solving

$$\min_{\{R_i^{(j)}\}} \sum_{i=1}^c \sum_{k < j} \left\| R_i^{(k)} D_i^{(k)} - R_i^{(j)} D_i^{(j)} \right\|_F^2 \quad (4)$$

$$\text{subject to: } R_i^{(j)T} R_i^{(j)} = I \quad \forall i, j.$$

Observe that Algorithm 2 produces an optimal solution with *zero* error, as there is always a perfect rotation between two sets of  $h$  orthogonal vectors. Therefore, one optimal solution of (4), which attains an objective value of zero, is to rotate all outlooks to a chosen final outlook. Namely, for  $m$  outlooks  $m - 1$  rotation matrices are computed for each class. Finally, shift the means of the rotated outlooks to those of the final outlook.

If we want to switch the choice of final outlook, all we need to do is apply the inverse mapping of the relevant outlook to all mapped outlooks. For example, to switch from outlook  $s$  to  $k$  one needs to apply the following transformation:

$$X_i^{(k)} = R_i^{(k)-1} \left( X_i^{(s)} - \hat{\mu}_i^{(s)} \right) + \hat{\mu}_i^{(k)} \quad \forall i.$$

## 5. Analysis

In this section we give a probabilistic robust interpretation of the rotation process, and prove a sample complexity bound on the convergence of the estimated rotation matrix.

### 5.1. Probabilistic Interpretation

In this section we discuss the effect of adding random noise to the utility matrices on the optimal rotation between two outlooks (Problem (2)). We do not assume knowledge of the probability distribution of the noise. Instead, we use its bounded total value for some chosen confidence level. We show that the solution to the noised problem is bounded by the sum of the solution to the original problem and a constant value that depends on the noise. Notably, the noise only has an additive effect to the bound.

Let  $\Delta$  be the additive random uncertainty to the utility matrix  $D_i^{(2)}$  for some class  $i$ . Suppose that

this uncertainty follows an unknown joint distribution  $\Delta \sim \mathcal{P}$ . This uncertainty may be portrayed by a chance-constrained extension of Problem (2)<sup>1</sup>:

$$\min_{R^T R = I, \tau} \tau \quad (5)$$

$$Pr_{\Delta \sim \mathcal{P}} \left\{ \left\| R(D^{(2)} + \Delta) - D^{(1)} \right\|_F \leq \tau \right\} \geq 1 - \eta,$$

where  $\eta \in [0, 1]$  is the desired confidence level.

Optimization of the chance constrained problem is natural, as it obtains, with high probability, the optimal rotation. However, despite their intuitive probabilistic form, chance constrained problems are generally intractable (Shapiro et al., 2009), thus we approximate Problem (5) as follows. We define  $\rho^* = \inf_{\alpha} \{Pr_{\Delta \sim \mathcal{P}} (\|\Delta\|_F \leq \alpha) \geq 1 - \eta\}$  and obtain that with probability at least  $1 - \eta$

$$\left\| R(D^{(2)} + \Delta) - D^{(1)} \right\|_F \leq \max_{\|\Delta\|_F \leq \rho^*} \left\| R(D^{(2)} + \Delta) - D^{(1)} \right\|_F.$$

Therefore, Problem (5) is upper bounded by the following minmax problem

$$\min_{R^T R = I} \max_{\|\Delta\|_F \leq \rho^*} \left\| R(D^{(2)} + \Delta) - D^{(1)} \right\|_F. \quad (6)$$

This is the *robust* version to the original rotation problem, with the uncertainty set  $\mathcal{U} = \{\Delta \mid \|\Delta\|_F \leq \rho^*\}$ <sup>2</sup>. Next, we construct the robust counterpart of (6).

**Theorem 1.** *Problem (6) is equivalent to*

$$\min_{R^T R = I} \left( \left\| R D^{(2)} - D^{(1)} \right\|_F \right) + \rho^*.$$

The proof is provided in Harel & Mannor (2011). The theorem shows that Problem (2) is robust to a perturbation of a total bounded value. That is, for a bounded noise, the only difference between the solution to the original problem and its robust version (Problem (6)) is an additive constant  $\rho^*$ . From a probabilistic point of view, the solution of this problem also provides a bound on the chance constrained problem in (5).

### 5.2. Sample complexity bounds

We next provide a bound for the sample complexity of the rotation step of the algorithm.

<sup>1</sup>Since Problem (2) is separable, the extension is done to each class separately. We drop the subscript  $i$ , representing the class, from the following derivations for brevity.

<sup>2</sup>The original rotation problem was actually the square of the Frobenius error. However, the two problems are equivalent since taking the square does not change the solution.

**Assumption 1.** (*Gaussian Mixture*) Each outlook is generated by a unique mixture of  $c$  Gaussian distributions, where  $c$  is the number of classes. The samples of each outlook are realizations of  $x \sim \sum_{i=1}^c w_i f_i(x)$ , where  $f_i(x) \sim \mathcal{N}(\mu_i, \Sigma_i)$  and  $\sum_{i=1}^c w_i = 1$ . We further assume that  $\|Exx^T\| \leq 1$  for each component.

**Theorem 2.** Suppose that Assumption 1 holds. For each outlook, let  $\delta, \epsilon_i, \epsilon \in (0, 1)$ , ( $i = 1, \dots, c$ ) and suppose that the number of samples for each class  $i$  satisfies:

$$n_i \geq C \frac{dh^2}{\epsilon_i^2} \log^2 \left( \frac{32dh^2}{\epsilon_i^2} \right) \log^2 \left( \frac{4hd}{\delta} \right).$$

Then

$$P \left( \left\| \hat{R} - R \right\| \leq \epsilon \right) \geq 1 - \delta,$$

where,  $\hat{R}$  is the estimated rotation matrix found by Algorithm 2,  $d$  is the dimension and  $C$  is a constant.

The proof of the theorem is provided in Harel & Mannor (2011). Note that the sample complexity of the mapping algorithm is dominated by the rotation stage. In practice, the number of chosen principal directions  $h$  is usually small. Also note that the bound on the norm of the second moment in Assumption 1 is achieved by the scaling stage.

## 6. Experiments

In this section we demonstrate our framework on activity recognition data, in which different users represent different outlooks. In this application, the multiple outlooks setup allows for valuable flexibility in real life recordings. For example, some users may use a simple sensor configuration for recordings, while others use a complex sensor board of multiple sensors. Also, this setup may resolve problems of varying sampling rates when using different hardware and workloads.

In our experiments we test two setups: a domain adaptation setup and a multiple outlook setup. For the domain adaptation setup a common feature representation is used, while for the multiple outlook setup a unique feature space is used for each user.

### 6.1. Data set description and feature extraction

The data set used for the experiments was collected by Subramanya et al. (2006) using a customized wearable sensor system. The system includes a 3-axis accelerometer, phototransistors for measuring light, barometric pressure sensors, and GPS data. The data consist of recordings from 6 participants who were asked to perform a variety of activities and record the labels. We used the following labels: walking, running, going upstairs, going downstairs and lingering.

After removing data with obvious annotation errors the data consists of about 50 hours of recording, divided approximately evenly among the 6 users. For each user the activities are roughly divided into 40% walking, 40 – 50% lingering, 2 – 5% running, 2 – 3% going upstairs, and 2 – 3% going downstairs. See (Subramanya et al., 2006) for further details on the sensor system and the recordings.

From the raw data we extracted windowed samples as follows. From the accelerometer data we used the x-axes measurements sampled at 512Hz, which we decimated to 32Hz. The barometric pressure sampled at 7.1Hz, was smoothed and interpolated to 32Hz. Next, we applied a two-second sliding window over each signal using a window of appropriate length. From each window a feature vector is extracted containing the Fourier coefficients of the accelerometer data, the mean of the gradient of the barometric pressure, and the mean values of the light signals. All together we obtained 20-35 thousand samples for each user with 37 features.

As explained in Section 3.2, before mapping the outlooks scaling should be applied to all the outlooks. For all the experiments, we scale the data to  $[0,1]$ . To reduce the sensitivity of the scaling to outliers we first collapse the extreme two percentile of the data to the value of the extreme remaining values (also known as Winsorization). Scaling parameters are chosen on the training data and applied to the test data. This preprocessing was applied to all baseline classifiers.

### 6.2. Domain Adaptation Setup

As mentioned above, multiple outlook learning may also be applied for domain adaptation. We tested both standard domain adaptation of two domains, as well as multiple source domain adaptation.

For the two domain problem we adopted the commonly used terminology in domain adaptation of *source* and *target* domains. We applied Algorithm 1 for different fractions of target labeled data and fully labeled source data. The performance was computed by 10-fold cross-validation, each fold containing random samples from each class according to its fraction in the complete set. The only parameter of the algorithm  $h$  was chosen on a random split.

We test the success of the mapping algorithm by classification of the target test data with a classifier trained on the mapped source data, denoted as the MOMAP classifier (no target data was used for training). This is a multi-class classification problem, with five possible labels. We use a multi-class SVM classifier with an

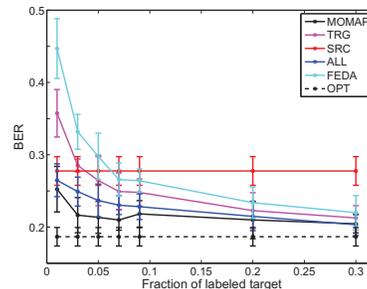
RBF-kernel ( $C = 64, \gamma = 0.25^3$ ) obtained by LIBSVM software (Chang & Lin, 2001). The data are unevenly distributed among the five classes, therefore we use the balanced error rate (BER) as a performance measure:  $BER = \frac{1}{c} \sum_{i=1}^c \frac{1}{n_i} e_i$ , where  $e_i$  and  $n_i$  are the numbers of errors and number of samples in class  $i$  respectively, and  $c$  is the number of classes.

We compare the MOMAP classifier to the following baselines: a target only classifier, trained on the available labeled target data (TRG); a source only classifier, trained on the source data (SRC); a classifier trained on all available labeled data of target and source (ALL); and the domain adaptation algorithm presented in (Daumé III, 2007) (FEDA). We also add the "optimal" error, obtained by training on the fully labeled target data (OPT).

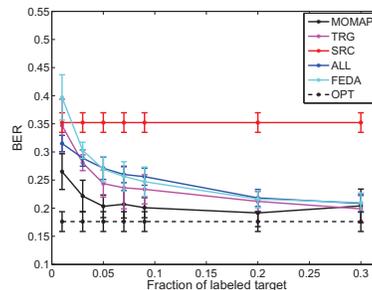
The results are presented in Figure 1. It can be observed that the MOMAP classifier outperforms the baseline classifiers for most fractions of target labeled data. The algorithm performs well across all sets of users, for example, for 5% labeled data it is significantly better (p-value < 0.05) than the TRG, SRC and FEDA classifiers for all sets, and significantly better than the ALL classifier for 18 out of 30 possible sets (see Table 1 in Harel & Mannor (2011)).

In the next experiment we consider mixtures of  $m$  source domains with some labeled data (both training and test sets are mixtures). We use the extension to multiple outlooks presented in Section 4 to find the mappings of the sources to each outlook. We test the classification performance on each component of the mixture with a classifier trained on all the mapped sources. The final performance measure is the mean BER averaged on all the sources. As in the previous experiment, the evaluation was done by 10-fold cross-validation, with the same classifier. The baselines are similar, with the change of the TRG to the mean value of multiple classifiers trained in each domain, and the ALL baseline to a classifier trained on all sources (the SRC classifier was not relevant). The experiment was performed on all 20 triplet combinations. Sample results are presented in Figure 2. These trends were consistent across users, for example, for 15% of labeled data the MOMAP algorithm outperforms all other classifiers for 15 of the combinations (p-value < 0.05). In the 5 remaining combinations, the algorithm performed significantly better than the TRG and FEDA algorithms, and equally well as the ALL classifier (see Table 2 in Harel & Mannor (2011)). For

<sup>3</sup>The parameters were chosen on the target classification problem. Common parameters were chosen for clear performance comparison of the different classifiers.



(a) User 5  $\rightarrow$  User 3



(b) User 6  $\rightarrow$  User 2

Figure 1. Domain adaptation setup for 2 domains.

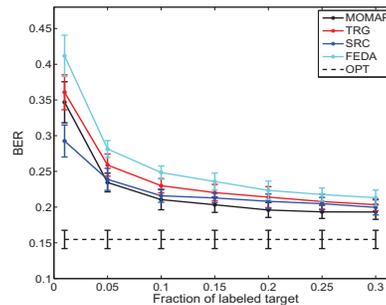


Figure 2. Domain adaptation setup for multiple outlooks: users 1,2 and 5.

larger portions of labeled data the MOMAP algorithm also obtained smaller error than the ALL classifier (p-value < 0.05). The effect of the ALL classifier may be a result of some regularization obtained from training on data from similar yet different domains.

### 6.3. Multiple Outlook Setup

We conducted three types of experiments for the multiple outlook setup, each with a different feature representation. The experiments' setup was similar to the previous experiments with some adjustments to the baselines: the SRC, ALL and FEDA baselines were no longer relevant, as the outlooks' features differ.

In the first experiment we tested the multiple outlook algorithm on two outlooks for the case of different sen-

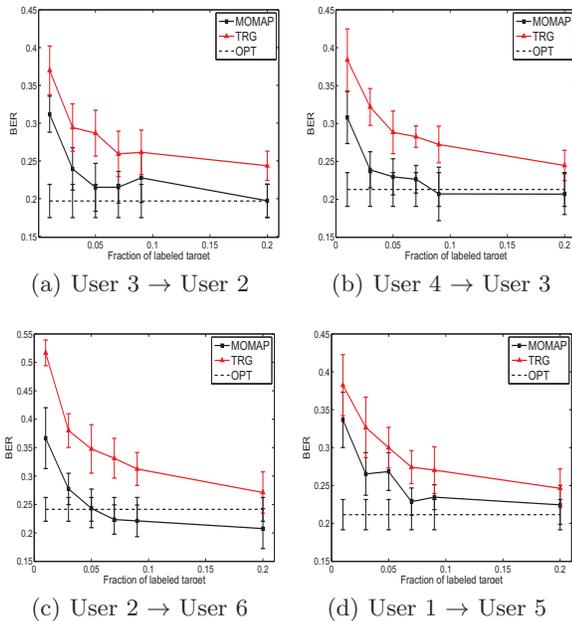


Figure 3. Two outlooks with different sensors. Final outlook: accelerometer and pressure. Mapped outlook: accelerometer, pressure and light sensors. The missing features in the final outlook are replaced by noise.

sors and added noise features. For the mapped outlook we used full feature representation (37 features). For the target outlook we used the accelerometer’s and pressure features, and excluded the light measurements. Instead of the light features we added features with Gaussian random noise ( $\mathcal{N}(0, 1)$ ). The experiment was performed on all pair combinations. For 5% labeled data of the learned outlook, the mean BER of the MOMAP was 4.5% ( $\pm 2.7\%$ ) lower than that of the TRG classifier. The results for four user pairs are presented in Figure 3. These results show that the mapping was successful, as training on the mapped data outperforms training on partial data in the target outlook. In Fig. 3(c) the MOMAP algorithm has lower error than the OPT classifier for some fractions; this may be a result of the added information in the light features.

In the second experiment we tried to learn from two outlooks with a different number of features resulting from different sampling rates. Specifically, for the learned outlook we kept the full feature representation as described in Section 6.1, while for the mapped outlook we used the same type of features but with 30Hz sampling rate instead of 32Hz. This resulted in 37 features in the target outlook and 35 in the mapped one. Note that our algorithm may be easily modified for this scenario; see Remark 1 in Section 3.2. For 5% labeled data the MOMAP algorithm had on aver-

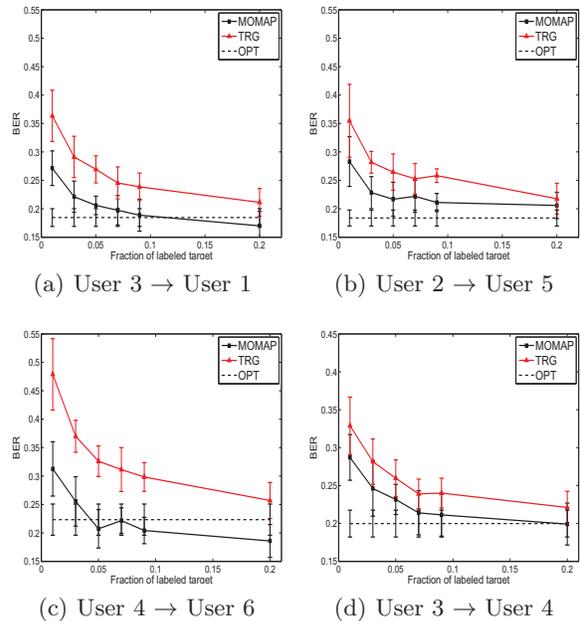


Figure 4. Multiple outlook learning for two outlooks with different sampling rates.

age 5.9% ( $\pm 2.4\%$ ) lower BER than the TRG classifier. Figure 4 presents the results on four user pairs. In Figs. 4(a) and 4(c) the MOMAP algorithm has lower error than the OPT classifier. Observe that this is possible since the balanced error rate is presented, which treats the error in different classes equally (namely, the MOMAP classifier does not outperform the non-balanced error).

In the third experiment we constructed the feature representation of each outlook from the 33 accelerometer’s features to which we added 10 features of Gaussian noise ( $\mathcal{N}(0, 1)$ ). We then randomly permuted the order of the features of each outlook. For this experiment, we used samples belonging to the walking, running and lingering classes, as we did not use the full feature set. The experiment was performed for two outlooks as well as for multiple outlooks. The results indicate the performance boost from MOMAP especially for the running activity. Due to space limitations we provide the results in Harel & Mannor (2011).

## 7. Future Work

Our proposed approach is a first step in developing the methodology for learning from multiple outlooks. This approach may be extended to many interesting directions. First, in this paper we only considered affine mappings between the outlooks and a natural extension is to consider richer classes of transformations such as piecewise linear mappings. Also, our ap-

proach is batch in the sense that first all the data have to be processed and then the classification algorithm can be used. A different extension of practical interest would be to develop an online version of the proposed approach that takes samples one by one and gradually improves the mapping. Finally, a major application domain, of independent interest, is natural language processing. Here the challenge would be to use a language where labels are abundant to better classify in a different language. The main obstacle here seems to be the nature of representation: language data are often represented as sparse vectors which may call for a different type of transformations between the outlooks.

## References

- Amini, M., Usunier, N., and Goutte, C. Learning from Multiple Partially Observed Views—an Application to Multilingual Text Categorization. In *Advances in Neural Information Processing Systems*, 2009.
- Blitzer, J., McDonald, R., and Pereira, F. Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 120–128. Association for Computational Linguistics, 2006. ISBN 1932432736.
- Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., and Wortman, J. Learning bounds for domain adaptation. In *Advances in Neural Information Processing Systems*, volume 20, pp. 129–136. Citeseer, 2007.
- Chang, C. and Lin, C. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Daumé III, H. Frustratingly Easy Domain Adaptation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics ACL*, volume 1, pp. 256–263. Association for Computational Linguistics, 2007.
- Gower, JC and Dijkstra, G.B. *Procrustes Problems*. Oxford University Press, USA, 2004.
- Ham, J., Lee, D., and Saul, L. Semisupervised alignment of manifolds. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, Z. Ghahramani and R. Cowell, Eds, volume 10, pp. 120–127. Citeseer, 2005.
- Harel, M. and Mannor, S. *Learning from Multiple Outlooks*, 2011. <http://arxiv.org/abs/1005.0027v1>.
- Hou, C., Zhang, C., Wu, Y., and Nie, F. Multiple view semi-supervised dimensionality reduction. *Pattern Recognition*, 43(3):720–730, 2010. ISSN 0031-3203.
- Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., and Scholkopf, B. Correcting sample selection bias by unlabeled data. In *Advances in Neural Information Processing Systems*, volume 19, pp. 601. Citeseer, 2007.
- Long, B., Yu, P.S., and Zhang, Z.M. A general model for multiple view unsupervised learning. In *Proceedings of the 8th SIAM International Conference on Data Mining (SDM'08), Atlanta, Georgia, USA*, 2008.
- Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, volume 21, pp. 1041–1048. Citeseer, 2009.
- Rudelson, M. and Vershynin, R. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM (JACM)*, 54(4):21, 2007.
- Rüping, S. and Scheffer, T. Learning with multiple views. In *Proceeding of the International Conference on Machine Learning Workshop on Learning with Multiple Views*, 2005.
- Satpal, S. and Sarawagi, S. Domain adaptation of conditional probability models via feature subsetting. In *Proceedings of Principles of Data Mining and Knowledge Discovery*, pp. 224–235. Springer, 2007.
- Shapiro, A., Dentcheva, D., Ruszczyński, A., and Ruszczyński, A.P. *Lectures on stochastic programming: modeling and theory*. Society for Industrial Mathematics, 2009. ISBN 089871687X.
- Shimodair, H. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90:227–244, 2000.
- Stewart, G.W and Sun, J.G. *Matrix Perturbation Theory*. Academic Press, 1990.
- Subramanya, A., Raj, A., Bilmes, J., and Fox, D. Recognizing activities and spatial context using wearable sensors. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. Citeseer, 2006.
- Wang, C. and Mahadevan, S. Manifold alignment using Procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 1120–1127. ACM, 2008.
- Wang, C. and Mahadevan, S. Manifold alignment without correspondence. In *Proceedings of the 21st International Joint Conferences on Artificial Intelligence*, 2009.